GET INTO PROGRAMMING WITH JavaScript

Axle Barr

Functions – the doing part of programming







Verbs - Action

Objects do not just sit and look pretty:

- Objects either do something or something can be done to the object
- The pot can be thrown, heated, held etc.
- The tire can support a car, roll, spin and so on
- The bird can whistle, walk, fly, eat and sleep
- The door can open, close and be knocked on

Functions

Function in the real world:

- Human body
- Vehicle
- Plants
- Computer
- HVAC

THE ACTIONABLE PART OF PROGRAMMING

Daily Functions

My Daily Functions:

Wake up

Brush teeth

Eat bf

Go to work

Return home

Go biking

shower

Think of all the functions you PERFORM in a single day

Daily Functions

My Daily Functions:

Wake up

Brush teeth

Eat bf

Go to work

Get sink fixed

Return home

Go biking

shower

THINK OF ALL THE FUNCTIONS YOU PERFORM IN A SINGLE DAY

Daily Functions

My Daily Functions:

Wake up

Brush teeth

Eat bf

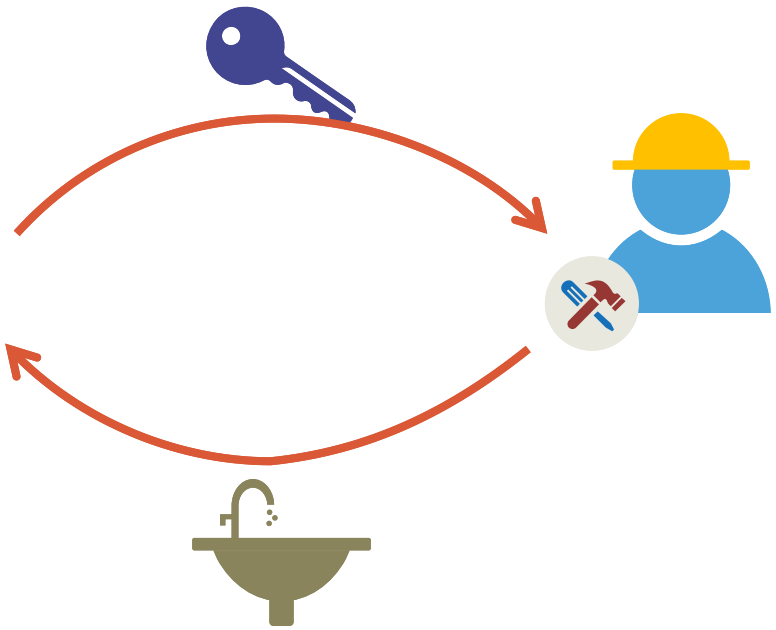
Go to work

getSinkFixed()

Return home

Go biking

shower



THINK OF ALL THE FUNCTIONS YOU PERFORM IN A SINGLE DAY

Coffee Maker

Coffee

Water

Coffee Maker

Function name

Parameters

makeCoffee(coffeeGrounds,water)

makeCoffee

Call the function (invoke)

FUNCTIONS AKA METHODS AKA SUB-ROUTINES

Facts about Functions

Blocks of code that perform a particular function

Input process output

The function will usually have a name and can be invoked anytime from any part of the program

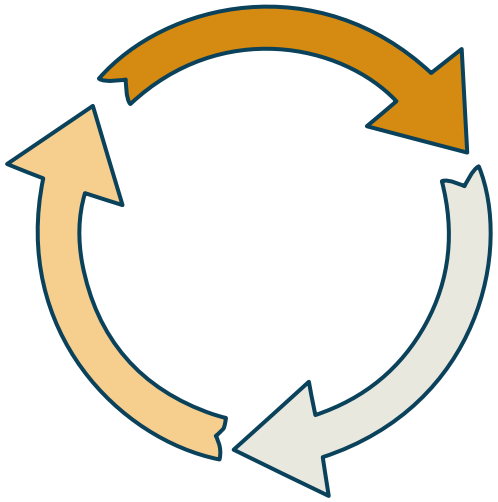
Any of the other programming structures can be part of a function

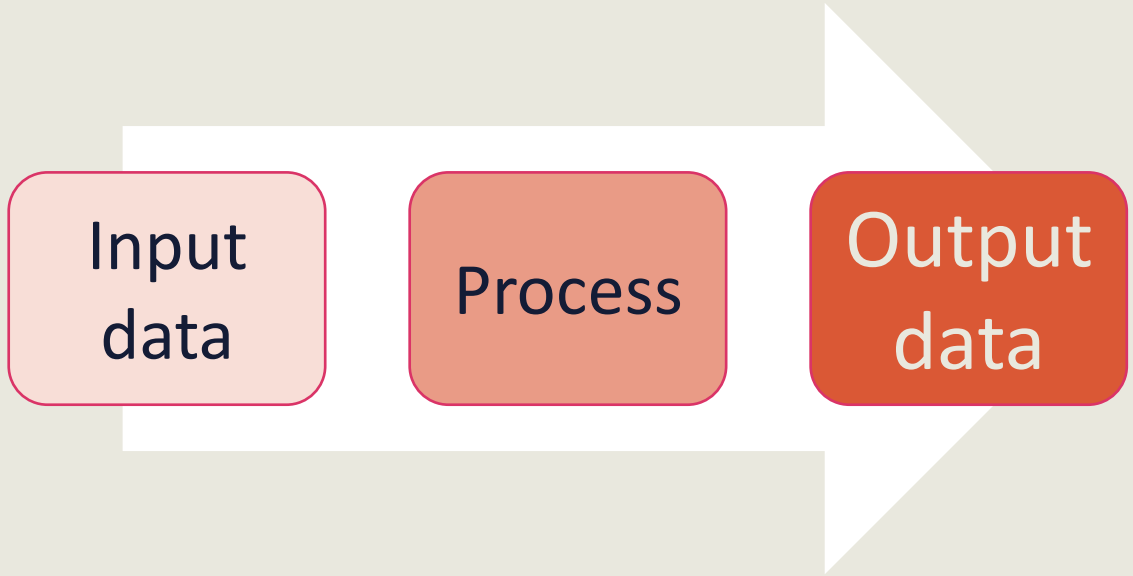
We can pass as many data points into the function, called **parameters**

Data or a result is passed back from the function, called the **return**

Functions are also called methods, sub-routines, modules

MINIATURE PROGRAMS





SIMPLE FUNCTION

```
function showOutput(){
  document.getElementById("js_output").innerHTML=“Hello”;
}
```

SIMPLE FUNCTION

function keyword, start of function

function name

```
function addEmUp(){
  3 + 4;
}
```

SIMPLE FUNCTION

```
function addEmUp(){
  3 + 4;
}
```

Bucket to pass other variable values to

Start of the function body

SIMPLE FUNCTION

```
function addEmUp(){
  3 + 4;
}
```

Function body

End of the function body

SIMPLE FUNCTION

```
function addEmUp(){
  3 + 4;
}

addEmUp();
```

Function call

SIMPLE FUNCTION

```
function addEmUp(x,y){
  x + y;
}

addEmUp();
```

Function requires parameters

SIMPLE FUNCTION

```
function addEmUp(x,y){
  x + y;
}

addEmUp( 4, 6 );
```

Function requires parameters

Pass two integers as parameters

SIMPLE FUNCTION

```
function addEmUp(x,y){
  return x + y;
}

addEmUp( 4, 6 );
```

Function returns a result to the caller

SIMPLE FUNCTION

```
function addEmUp(x,y){
  return x + y;
}
```

Function returns a result to the caller

```
sum = addEmUp( 4, 6 );
```

Function returns a result to the caller

SIMPLE FUNCTION

```
let sum = 0;

function addEmUp(x,y){
  return x + y;
}


sum = addEmUp( 4, 6 );
```

Declare and initialize *sum*

SIMPLE FUNCTION

```
let sum = 0;

function addEmUp(x,y){
  return x + y;
}


sum = addEmUp( 4, 6 );

printOut = sum;
```

MORE COMPLEX FUNCTIONS

```
let moreProducts = true, productCost=0.0, totalCart=0.0, moreItems=false;
while (moreProducts == true) {
        productCost = prompt("Enter price of product: ");
        productCost = parseFloat(productCost);
        totalCart = totalCart + productCost;
        moreItems = confirm("Do you have more items?");
        if(moreItems == false){
                moreProducts = false;
        }
}
```

The ***else*** part is missing!

MORE COMPLEX FUNCTIONS

```
let moreProducts = true, productCost=0.0, totalCart=0.0, moreItems=false;
function getCartTotal ( ) {
        while (moreProducts == true) {
                productCost = prompt("Enter price of product: ");
                productCost = parseFloat(productCost);
                totalCart = totalCart + productCost;
                moreItems = confirm("Do you have more items?");
                if(moreItems == false){
                        moreProducts = false;
                }
        }
}
```

MORE COMPLEX FUNCTIONS

```
let moreProducts = true, productCost=0.0, totalCart=0.0, moreItems=false;
function getCartTotal ( ) {
        while (moreProducts == true) {
                productCost = prompt("Enter price of product: ");
                productCost = parseFloat(productCost);
                totalCart = totalCart + productCost;
                moreItems = confirm("Do you have more items?");
                if(moreItems == false){
                        moreProducts = false;
                }
        }
        return totalCart;
}
```

The purpose of this function is to give the calling function a total

```
function showOutput(){
 printOut = getCartTotal();
 document.getElementById("js_output").innerHTML=printOut;
}
```

```
let moreProducts = true, productCost=0.0, totalCart=0.0, moreItems=false;
function getCartTotal ( ) {
        while (moreProducts == true) {
                productCost = prompt("Enter price of product: ");
                productCost = parseFloat(productCost);
                totalCart = totalCart + productCost;
                moreItems = confirm("Do you have more items?");
                if(moreItems == false){
                        moreProducts = false;
                }
        }
        return totalCart;
}
```

Multiple Functions

```
function getCartTotal(){
  while (moreProducts == true) {
    productCost = prompt("Enter price of product: ");
    productCost = parseFloat(productCost);
    totalCart = totalCart + productCost;
    moreItems = confirm("Do you have more items?");
    if(moreItems == false){
      moreProducts = false;
    }
  }
  return totalCart;
}
```

Multiple Functions

```
function getCartTotal(){
  while (moreProducts == true) {
    productCost = prompt("Enter price of product: ");
    productCost = parseFloat(productCost);
    totalCart = totalCart + productCost;
    moreItems = confirm("Do you have more items?");
    if(moreItems == false){
      moreProducts = false;
    }
  }
  return totalCart;
}
```

```
function calculateTaxes(){
  let afterTax = 0.0;
  let beforeTax = getCartTotal();
  afterTax = beforeTax * 1.08;
  return afterTax;
}
```

Multiple Functions

```
function getCartTotal(){
  while (moreProducts == true) {
    productCost = prompt("Enter price of product: ");
    productCost = parseFloat(productCost);
    totalCart = totalCart + productCost;
    moreItems = confirm("Do you have more items?");
    if(moreItems == false){
      moreProducts = false;
    }
  }
  return totalCart;
}
```

```
function calculateTaxes(){
  let afterTax = 0.0;
  let beforeTax = getCartTotal();
  afterTax = beforeTax * 1.08;
  return afterTax;
}
```

```
function showOutput(){
  printOut = calculateTaxes();

document.getElementById("js_out
put").innerHTML=printOut;
```

Multiple Functions

```
function getCartTotal(){
  while (moreProducts == true) {
    productCost = prompt("Enter price of product: ");
    productCost = parseFloat(productCost);
    totalCart = totalCart + productCost;
    moreItems = confirm("Do you have more items?");
    if(moreItems == false){
      moreProducts = false;
    }
  }
  return totalCart;
}
```

```
function calculateTaxes(){
  let afterTax = 0.0;
  let beforeTax = getCartTotal();
  afterTax = beforeTax * 1.08;
  return afterTax;
}
```

```
function showOutput(){
  printOut = calculateTaxes();

document.getElementById("js_out
put").innerHTML=printOut;
```

Multiple Functions

```
function getCartTotal(){
  while (moreProducts == true) {
    productCost = prompt("Enter price of product: ");
    productCost = parseFloat(productCost);
    totalCart = totalCart + productCost;
    moreItems = confirm("Do you have more items?");
    if(moreItems == false){
      moreProducts = false;
    }
  }
  return totalCart;
}
```

```
function calculateTaxes(){
  let afterTax = 0.0;
  let beforeTax = getCartTotal();
  afterTax = beforeTax * 1.08;
  return afterTax;
}
```

```
function showOutput(){
  printOut = calculateTaxes();

document.getElementById("js_out
put").innerHTML=printOut;
```

Multiple Functions

```
function getCartTotal(){
  while (moreProducts == true) {
    productCost = prompt("Enter price of product: ");
    productCost = parseFloat(productCost);
    totalCart = totalCart + productCost;
    moreItems = confirm("Do you have more items?");
    if(moreItems == false){
      moreProducts = false;
    }
  }
  return totalCart;
}
```

```
function calculateTaxes(){
  let afterTax = 0.0;
  let beforeTax = getCartTotal();
  afterTax = beforeTax * 1.08;
  return afterTax;
}
```

```
function calculateShipping(){
  let shipCharge = 0.0, cartTotal = 0.0;
  cartTotal = getCartTotal();
  if(cartTotal < 100)
    shipCharge = 5.0;
  return shipCharge;
}
```

```
function showOutput(){
  printOut = calculateTaxes() +   calculateShipping();
  document.getElementById("js_output").innerHTML=printOut;
}
```

FLOWCHART TO SHOW FUNCTION/SUBROUTINE